CLAIMS

We claim:

Sub B1

~~1. A method of monitoring processor resources, said method comprising:~~

~~determining a set of needed resources for a block of code;~~

~~testing if said set of resources are available at a start of said block of code; and~~

~~signaling an error if said set of resources needed for said block of code are not available.~~

~~2. The method as claimed in claim 1, said method further comprising:~~

~~determining a set of available resources that will be available after said block of code has executed.~~

3. The ~~method as claimed~~ in claim 1 wherein said needed resources comprise stack contents

Sub B2

4. The method ~~as claimed~~ in claim 1 wherein said set of needed resources is determined at a compile time.

Sub B1

~~5. The method as claimed in claim 1 wherein said set of needed resources is determined dynamically.~~

~~6. The method as claimed in claim 1 wherein signaling said error if said set of resources needed for said block of code are not available comprises branching to a fault handler routine.~~

~~7. The method as claimed in claim 6 wherein signaling said fault handler routine simulates a processor exception.~~

~~8. The method as claimed in claim 1 wherein needed resources are represented by a bit vector.~~

~~9. The method as claimed in claim 8 wherein said bit vector is generated dynamically.~~

Sub B3 10. A computer-readable medium having stored thereon a set of instructions to monitor processor resources, said set of instruction, which when executed by a processor, cause said processor to perform a method comprising:

determining a set of needed resources for a block of code;

testing if said set of resources are available at a start of said block of code; and

signaling an error if said set of resources needed for said block of code are not available.

11. The computer-readable medium as claimed in claim 10, wherein said set of instructions further includes additional instructions, which when executed by said processor, cause said processor to perform said method further comprising:

determining a set of available resources that will be available after said block of code has executed.

12. The computer-readable medium as claimed in claim 10 wherein said needed resources comprise stack contents .

Sub Bi 13. The computer-readable medium as claimed in claim 10 wherein said set of needed resources is determined at a compile time.

Sub B1

14. The computer-readable medium as claimed in claim 10 wherein said set of needed resources is determined dynamically.

15. The computer-readable medium as claimed in claim 10 wherein signaling said error if said set of resources needed for said block of code are not available comprises branching to a fault handler routine.

16. The computer-readable medium as claimed in claim 15 wherein signaling said fault handler routine simulates a processor exception.

17. The computer-readable medium as claimed in claim 10 wherein needed resources are represented by a bit vector.

18. The computer-readable medium as claimed in claim 17 wherein said bit vector is generated dynamically.

Sub B'

19. A computer-readable medium, having stored thereon a first set of instructions, the first set of instructions, which when executed by a processor, generate a second set of instructions through a binary translation process, the second set of instructions when executed by the processor, cause said processor to perform a method comprising:

determining a set of needed resources for a block of code;

testing if said set of resources are available at a start of said block of code; and

signaling an error if said set of resources needed for said block of code are not available.

20. The computer-readable medium as claimed in claim 19, wherein said set of instructions further includes additional instructions, which when executed by said processor, cause said processor to perform said method further comprising:

determining a set of available resources that will be available after said block of code has executed.

21. The computer-readable medium as claimed in claim 19 wherein said needed resources comprise stack contents .

Sub B1

22. The computer-readable medium as claimed in claim 19 wherein said set of needed resources is determined dynamically.

23. The computer-readable medium as claimed in claim 19 wherein signaling said error if said set of resources needed for said block of code are not available comprises branching to a fault handler routine.

24. The computer-readable medium as claimed in claim 23 wherein signaling said fault handler routine simulates a processor exception.

25. The computer-readable medium as claimed in claim 19 wherein needed resources are represented by a bit vector.